

2
/PAL III
/PART III
/PAGE 19

*1000

/ROUTINE TO PROCESS SPECIAL CHARACTERS

/CALLED BY ROUTINE NCHEK WHEN IT DISCOVERS A SPECIAL CHARACTER

/SPSHUL ASSUMES THAT THE SPECIAL CHARACTER DELIMITS A SYMBOL OR OCTAL WORD

/A SEARCH IS MADE THROUGH THE SYMBOL TABLE FOR THE SYMBOL, IF ANY

/THE VALUE OF THE SYMBOL, OR OCTAL WORD, IS LEFT IN REGISTER SYMVAL

/THE SYMBOL TYPE IS LEFT IN REGISTER TYPE

/CODES FOR SYMBOL TYPES ARE AS FOLLOWS:

/MEMORY REF INSTRUCTIONS 0

/USER SYMBOLS& MICRO OPS +1

/LENGTH: 85 REGISTERS

/PAGE 20

SPSHUL, CLA

/JUNK IN AC WHEN SPSHUL IS CALLED

TAD SYM1

/SYMBOL TO HANDLE?

SNA CLA

JMP OCTALW

/NO, GO PROCESS OCTAL WORD

JMS I SMGETI

/SEARCH FOR SYMBOL

JMP PUTIN

/NOT THERE, PUT IT IN THE SYMBOL TABLE

CLL

TAD OPCDP

/IS IT A PSEUDO-CODE?

SNI

JMP PSEUDA

/YES: PROCESS IT

TAD MICRP

/NO: IS IT A MICRO INSTR.?

SNI CLA

IAC

/YES: CODE=+1

PUSHSP, DCA TYPE

/STORE CODE IN REGISTER TYPE

TAD CHAR

/GET THE SPECIAL CHARACTER

TAD M211

/IS IT A TAB?

SNA

JMP I SPACI

/YES: TREAT AS A SPACE

TAD M4

/IS IT A CARRIAGE RETURN?

SNA

JMP I CARRI

/YES, PROCESS CARRIAGE RETURN

TAD M23

/IS IT A SPACE (CODE 240)?

SNA

JMP I SPACI

/YES: PROCESS SPACE

TAD M4

/IS IT A DOLLAR SIGN (CODE 244)?

SNA

JMP I DOLLI

/YES, PROCESS \$

TAD M6

/IS IT A STAR (CODE 252)?

SNA

JMP I STARI

/YES, PROCESS STAR

TAD M1

/IS IT PLUS (CODE 253)?

SNA

JMP I ADDI

/YES, PROCESS PLUS

TAD M1

/IS IT A COMMA (CODE 254)?

SNA

JMP I SYMSTI

/YES, PROCESS COMMA

TAD M1

/IS IT A MINUS (CODE 255)?

SNA

JMP I SUBI

/YES, PROCESS MINUS

TAD M1

/IS IT A PERIOD (CODE 256)?

SNA

JMP I POINTI

/YES, PROCESS PERIOD

TAD M17

/IS IT AN = (CODE 275)?

SNA

JMP I EQUALI

/YES, PROCESS EQUAL SIGN

TAD C2

/IS IT A SEMI-COLON?

SNA CLA

JMP I CARRI

/YES: TREAT AS CR

/PAGE 21

ERRORN, TAD CHAR

/NO, ILLEGAL CHARACTER

PUSH

/SAVE IT

TAD C4

/CODE FOR IC IS 4

PUSHJ

/CALL ERROR

ERROR2

JMS I AAA

/GET ANOTHER

JMP I ,+1

/RETURN TO NCHEK

NCHEK+1

PSEUDA, CLA

/CLEAR INFO FOR ALL PSEUDO-OPS

DCA INFO

TAD SYMVAL

/GET ADDRESS OF ROUTINE

DCA ,+2

PUSHJ

/CALL ROUTINE

0

JMP PUSHSP

/HANDLE TERMINATOR

PUTIN,

TAD M1

/STORE SYMBOL WITH AUTO=INDEXC

DCA AUTOA

/ADDRESS-1 GOES INTO AUTOA

TAD SYM1

/FIRST WORD OF SYMBOL

TAD C4TH

/PUT IN UNDEFINED BIT

DCA I AUTOA

TAD SYM2

/SECOND WORD OF SYMBOL

DCA I AUTOA

TAD SYM3

/THIRD WORD OF SYMBOL

DCA I AUTOA

TAD AACT

/VALUE OF SYMBOL

DCA I AUTOA

TAD C4

/MOVE S.T. POINTER BACK 4 REGISTERS

TAD SPNT

DCA SPNT

JMP PUSHSP-1

OCTALW,

TAD OCWD

/GET OCTAL WORD

DCA SYMVAL

/SAVE IT AS THE SYMBOL VALUE

DCA OCWD

JMP PUSHSP-1

/PROCESS AS USER SYMBOL

MICRP, OPCD=MICR

OPCDP, -OPCD

SMGETI, SYMGET

```

/PAGE 22
/SYMBOL COMPARISON ROUTINE
/COMPARES SYMBOLS: ADDRESSES IN INDEX REGISTERS
/SORT1 AND SORT2.
/THE ROUTINE COMPARES A(I) WITH B(I)
/EXITS TO CALLING ADDRESS+1 IF A=B
/EXITS TO CALLING ADDRESS+2 IF A>B
/EXITS TO CALLING ADDRESS+3 IF A<B
/WHERE A IS SYMBOL IN SORT1, B IS SYMBOL IN SORT2
/LENGTH: 30 REGISTERS

```

SYMCOM, 0

```

      CLL                      /CLEAR LINK
      TAD I SORT1
      AND MASK                  /DELETE DEFINED BIT
      DCA POPTM
      TAD I SORT2
      AND MASK
      CIA
      TAD POPTM
      SZA CLA                    /FIRST PAIR EQUAL?
      JMP TSIGN=2
      CLL
      TAD I SORT2              /SECOND WORD OF SYMBOL 1
      CMA IAC
      TAD I SORT1              /=SECOND WORD OF SYMBOL 2
      SZA CLA                    /SECOND PAIR EQUAL?
      JMP TSIGN=1              /NO: TEST SIGN
      CLL
      TAD I SORT2              /THIRD WORD OF SYMBOL 1
      CMA IAC
      TAD I SORT1              /=THIRD WORD IF SYMBOL 2
      SNA CLA
      JMP C5OUT                /ALL THREE ARE EQUAL
      JMP TSIGN
      ISZ SORT2
      ISZ SORT2
TSIGN, SNL                      /TEST SIGN
C3OUT, ISZ SYMCOM              /EXIT IF SYMBOL1<SYMBOL2
C4OUT, ISZ SYMCOM              /EXIT IF SYMBOL1>SYMBOL2
C5OUT, JMP I SYMCOM            /EXIT IF SYMBOL1=SYMBOL1

```

/PAGE 23
/PUSHJ ROUTINE TO RECURSIVELY CALL SUBROUTINES
/CHECKS FOR PUSHDOWN OVERFLOW VIA THE
/PUSH ROUTINE
/LENGTH: 9 REGISTERS

PUSHJM, 0

DCA ITEM	/CARRY C(AC) TO SUBROUTINE
TAD I PUSHJM	/ADDRESS OF WHERE TO JUMP
DCA POPTM	/SAVE IT
TAD PUSHJM	/GET RETURN
IAC	
PUSH	/PUSH DOWN
TAD ITEM	/RESTORE C(AC)
JMP I POPTM	/CALL SUBROUTINE

L10,

/PAGE 24

/ALPHABETIC OUTPUT ROUTINE

/CALLED BY JMS I EEE, WITH PACKED SIXBIT WORD IN AC

/A CALL WITH 0 IN THE AC, WILL GENERATE TWO SPACES

/LENGTH: 35 REGISTERS

*1200

ALFO,	0	
	DCA ALFTEM	/SAVE THE PACKED CHARACTER
	TAD M2	/PUT -2 IN THE COUNTER - 2 WORD TO
	DCA ALFTEM+1	/BE OUTPUTTED
	TAD ALFTEM	/GET FIRST CHARACTER
	RTR	/ROTATE INTO POSITION
	RTR	
	RTR	
NXCL,	AND SL6	/MASK OFF EXTRANEQUS BITS
	SNA	/IS THE CHARACTER A ZERO?
	JMP ALFL-1	/YES, OUTPUT A SPACE
	TAD M43	/CHECK FOR LINE FEED -43
	SNA	
	JMP FLF	/GO TO SPECIAL LINE FEED SECTION
	TAD M2	/CHECK FOR CARRIAGE RETURN -45
	SNA	
	JMP FCR	/GO TO SPECIAL CR SECTION
	TAD C5	/CALCULATE CHAR -40
	SPA	/IF RESULT IS .GE. 0, ADD 240
	TAD C100	/OTHERWISE, ADD 340
	TAD C240	
ALFL,	JMS I DDD	/GO TO TYPEOUT ROUTINE
NEXL,	TAD ALFTEM	/PICK UP PACKED WORD
	ISZ ALFTEM+1	/ALL DONE?
	JMP NXCL	/NO, GO PROCESS IT
	CLA	/YES, EXIT WITH CLEARED AC
	JMP I ALFO	/EXIT
FLF,	TAD M3	/GET LF CODE(212)
FCR,	TAD C215	/GET THE CODE FOR CARRIAGE RETURN
	JMP ALFL	/GO OUTPUT IT
M43,	-0043	
C100,	0100	
C240,	0240	
ALFTEM,	0	
	0	

/PAGE 25

/OCTAL OUTPUT ROUTINE

/CALLED WITH JMS I OCTI, OCTAL NUMBER IN THE ACCUMULATOR

/LENGTH: 17 REGISTERS

```
OCTO,    0
          DCA ALFTEM          /SAVE WORD
          TAD M4              /COUNTER FOR 4 CHARACTERS
          DCA ALFTEM+1
NEXT2,    TAD ALFTEM          /PICK UP OUTPUT WORD
          RAL                 /ROTATE THREE BITS LEFT
          RTL
          DCA ALFTEM          /SAVE RESULTS
          TAD ALFTEM          /PICK IT UP AGAIN
          RAL                 /GET LAST BIT FROM LINK
          AND SL3              /TRIM TO 3 BITS
          TAD C260             /CONVERT TO ASCII
          JMS I DDD            /OUTPUT IT
          ISZ ALFTEM+1         /DONE 4?
          JMP NEXT2           /NO
          JMP I OCTO          /YES, EXIT

C260,     0260
```

/ROUTINE TO GENERATE LEADER-TRAILER

/LENGTH: 19 REGISTERS

```
LDTR,     0
          TAD KONS              /SET COUNTER FOR
          DCA TEM2              /# OF CODES
LD1,       TAD PF                /WHICH PASS IS THIS?
          SZA SMA CLA
          JMP .+5                /PASS 2: USE PUNCH
          CLA OSR                /PASS 1 OR PASS 3
          RAR                    /TEST BIT 11
          SNL CLA
          JMP .+4                /0: USE 33ASR
          TAD C200               /1: USE PUNCH
          JMS I BBB              /PUNCH IT
          JMP .+3
          TAD C200
          JMS I DDD              /33 ASR
          ISZ TEM2              /DONE?
          JMP LD1                /NO: CONTINUE
          JMP I LDTR             /YES: EXIT ROUTINE
KONS,      -100                 / L/T CONSTANT
```

/PAGE 26

/BINARY PUNCH ROUTINE

/LINK = 1 IF ORIGIN SETTING

/CALLED WITH JMS I PUNI, BINARY WORD IN ACCUMULATOR

/LENGTH: 22 REGISTERS

```
BINP, 0
      DCA TEM1          /SAVE WORD TO BE PUNCHED
      TAD TEM1          /LINK IS ONE IF ORIGIN
      RTR               /PUT CHARACTER INTO POSITION
      RTR
      RTR
      AND SL7           /SAVE SEVEN BITS
      DCA TEM3          /SAVE IT
      TAD TEM3          /PICK IT UP
      JMS I BBB         /AND GO TO PUNCH ROUTINE
      TAD CKSM          /UPDATE CHECKSUM
      TAD TEM3
      DCA CKSM
      TAD TEM1          /PICK UP ORIGINAL CHARACTER AGAIN
      AND SL6           /SAVE LOW ORDER SIX BITS
      DCA TEM3          /SAVE IT FOR CHECKSUM
      TAD TEM3          /PICK IT UP
      JMS I BBB         /PUNCH IT
      TAD CKSM          /UPDATE CHECKSUM
      TAD TEM3
      DCA CKSM
      JMP I BINP        /EXIT
```

/PAGE 27
 /ROUTINE TO PLACE ITEMS ON PUSHDOWN LIST
 /CHECKS FOR PUSHDOWN OVERFLOW
 /LENGTH: 17 REGISTERS

PUSHIN, 0

DCA I PSHPN1 /SAVE AC ON PUSHDOWN LIST
 ISZ PSHPN1 /INCREMENT PUSHDOWN POINTER
 TAD PIEND /CHECK FOR PUSHDOWN OVERFLOW
 TAD PSHPN1 /HAS POINTER PASSED PIEND?
 SPA CLA
 JMP I PUSHIN /NO, POINTER IS OK, EXIT
 TAD PSHPN1 /GET ADDRESS OF PUSHDOWN LIST FOR ERROR
 DCA I PSHPN1 /SAVE IT ON LIST FOR ERROR PICKUP
 ISZ PSHPN1 /INCREMENT PUSHDOWN POINTER
 TAD PERR1 /GET RETURN ADDRESS FOR ERROR CALL
 DCA I PSHPN1 /THUS SIMULATING A PUSHJ
 JMP I PERR3 /HOP INTO THE ERROR ROUTINE

PERR1, .+1 /ERROR RETURN ADDRESS
 HLT /HALT ON ERROR RETURN
 JMP .-1 /NO RECOVERY

PERR3, ERROR2 /ERROR ADDRESS FOR SIMULATED PUSHJ

/ROUTINE TO ASSEMBLE OCTAL AND DECIMAL
 /NUMBERS ; SWITCH IS MADE BY OCTAL AND DECIMA
 /PSEUDO OPS
 /LENGTH: 17 REGISTERS

NMBR, ISZ INFO /VALID INFORMATION
 NEXNUM, TAD OCWD /GET CURRENT WORD

CLL RAL
 CLL RAL

NSWCH, NOP /SWITCH FOR OCTAL/DECIMA

TAD OCWD
 CLL RAL
 DCA OCWD /STORE OCWD*8 OR OCWD*10
 TAD CHAR
 TAD M60 /TURN INTO DIGIT
 TAD OCWD /ADD TO CURRENT
 DCA OCWD

JMS I AAA /FETCH A CHARACTER
 JMS I NCHEK1 /CHECK IT
 JMP NEXNUM /ALL VALID NUMBERS
 JMP I RONGI /LETTER IN WORD BEGUN WITH NUMBER

M60, -60
 RONGI, ERRORN /CALL ERROR-COMBINATION NOT LEGAL
 L12,

PAUSE